IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	: Scholz et al	Art Unit	: 2121
Serial No.	: 10/698,956	Examiner	: Darrin D. Dunn
Filed	: October 31, 2003	Conf. No.	: 8804
Title	: BLOCKING INPUT WITH DELAYED MESSAGE		

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION UNDER RULE 1.132

1. I am an inventor of the patent application mentioned above ("the present patent application").

2. I know that the United States Patent and Trademark Office ("PTO"), in its examination of the present patent application, has currently rejected the claims thereof based on the following three United States Patent Application Publications:

   U.S. 20070250840 ("Coker")
   U.S. 20040088700 ("Lee")
   U.S. 20040103174 ("Balducci")

3. I have been informed that with regard to the independent claims, the PTO is citing Lee as disclosing "a system for automatically installing software on a client via a server," and for this reason I have not reviewed this reference in detail. I merely note that the software Lee installs on the client appears designed for a client device. That is, Lee does not suggest that server-oriented software should be installed in a client environment.

4. I have reviewed the Coker and Balducci references. Beginning with Coker, it is entitled "Computing system and method to perform run-time extension for world wide web application". In the abstract, Coker explains that the run-time extension is done "to

modify the functionality of a corresponding World Wide Web (WWW) application object". Coker abstract. In the description, Coker introduces run-time extension with regard to a client-server system configuration, and FIG. 2 shows such a system including, among other components, a web client 205, a mobile web client 210 and an enterprise server 250. Coker's FIGS. 5A and 5B, moreover, show a "system framework or infrastructure 500 to support an interactive web client 205 and a mobile web client 210 of FIG. 2 ... [and] an alternative view of the exemplary system framework or infrastructure 500," respectively. Coker [0108]. FIG. 5C, in turn, shows a "computing system to extend a Web application". Coker [0112]. I will discuss some aspects of Coker's extension of application objects.

5. With reference to FIG. 5C, Coker describes that a client 502 includes applets, or objects, that are proxies of applets on a server 504. Coker [0112-13]. Coker says that "functionality of the applets [on the server] can be extended by modifying the source codes for the applets" and that "source code that extends a class or an object can be input by the user through the user interactive display 580". Coker [0114]. Coker says that when an object thus modified on the server device needs to be updated on the client device, "[s]ynchronization can be accomplished through a remote procedure call (RPC) mechanism 528 and a notification mechanism 530." Coker [0162]. Thus, Coker does provide a synchronization procedure between objects on separate devices.

6. The notification mechanism 530 used in the synchronization, moreover, also can be used by the server device in alerting the client device about a possible delay. Particularly, Coker describes a busy-state manager with reference to FIG. 42. Coker describes that "when a client submits a request to a server that involves long-running or time-consuming operations on the server side, the server can inform the client accordingly using the notifications mechanism". Coker [0307-8]. That is, in Coker, it is the server that notifies the client about possible delays in processing a request. It is also my

understanding that the request in Coker is tied to the particular client application that makes the request.

7. Not only is the notification from the busy state manager in Coker sent from the server to the client, it also is based on a prediction that a request may be delayed, not on any monitoring to determine whether a delay actually occurs. For example, Coker describes that the busy-state manager can determine "whether the request *may* involve long-running server operations or *may* take a long time to finish". Coker [0308] (emphasis added). Coker's predictive approach does not determine whether any client request actually takes longer than a specific time to complete. Moreover, Coker's server-based approach can lead to user inconvenience because the server does not predict delays that are due to poor network connection between the server and the client (and even if such a delay were predicted, which Coker does not do, the notification from the server may take too long to reach the client).

8. It also appears that Coker does not lock the client device until some time after the request is transmitted. Coker states that "*once the client is informed by the server* that the request may take a long time to process, the client can inform the user that the request processing has started and lock the user interface during the time the request is being processed by the server". Coker [0309] (emphasis added). This is also reflected in Coker's flow chart in FIG. 43, where the client sends the request in a step 4310, and does not lock the user interface until a step 4340, after the server has informed the client that the request *may* involve long-running operations. Coker [0310]. Thus, Coker does not lock the client device for the duration of the client-server communication, but only for a portion thereof.

9. One of the approaches Coker describes is that the client device alerts its user that there is a possible delay, for example by announcing that "the request processing has started and *may take a long time to finish*". Coker [0310] (emphasis added). Another approach that Coker describes for informing the user is to display a progress bar showing how much of

the request the server has completed. Coker [0310]. Both these approaches of informing the user (after the server has predicted a slow response) may be distracting to the user because they do not take into account whether completion of the current request is actually delayed. *See, e.g.*, page 1, lines 26-29 in the specification of the present patent application. Note that Coker's monitoring of the server's progress in completing a client request does not indicate whether the server is taking too long to finish the request.

10. In summary regarding Coker, I note that this reference relies on an approach where the server device notifies the client device, and only about predicted delays; and that in such situations the client presents the notification to the user regardless of actual delay. Moreover, the locking of the client device is conditioned on first receiving the (predicted) delay notification from the server.

11. Balducci, in turn, is entitled "Folder synchronization" and describes "synchronizing folders between a mobile device and a second computing device". Balducci abstract. To that end, Balducci describes a mobile device 12 and a computing device 14 with reference to FIG. 1. Balducci [0022]. Particularly, Balducci uses synchronization engines 24 and 36 in the respective devices to manage synchronization of object stores for two applications exemplified as MICROSOFT OUTLOOK and MICROSOFT WORD. Balducci [0024-27].

12. But Balducci's synchronization engines do not necessarily monitor all communications between the two devices. This is because synchronization is only necessary if an object on either device has been changed. Particularly, Balducci describes that *"[w]hen a user changes one instance of the object* stored in either object store 22 or 34, the second instance of that object in the other of stores 22 and 34 is out of sync and is desirably updated the next time mobile device 12 has two-way communication with computing device 14". Balducci [0027] (emphasis added). I understand this to mean that not every

communication between Balducci's mobile device and the second computing device involves the synchronization engines.

13. Finally, I note that both Balducci and Coker perform synchronization. As I have discussed above, Coker uses the RPC mechanism 528 and the notification mechanism 530 in this regard, while Balducci, in contrast, uses the synchronization engines 24 and 36. If one were to hypothetically substitute Balducci's synchronization engines in Coker's system, then the notification mechanism 530 (and possibly also the RPC mechanism 528) would not be needed for synchronization purposes anymore. Also, I reiterate that Coker explicitly described how the busy-state manager makes use of the notification mechanism 530. Coker [0307-8].

14. I hereby declare that all statements made herein of my knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patents issued thereon.

Signed: Martin Scholz

Date: 31.07.09